

WHAT IS CLAIMED IS:

1 1. A computer system, comprising:

2 an interconnect;

3 a plurality of processor nodes, coupled to the interconnect, each processor node

4 including:

5 at least one processor core, each processor core having an associated memory
6 cache for caching memory lines of information;

7 an interface to a local memory subsystem, the local memory subsystem storing
8 a multiplicity of memory lines of information; and

9 a protocol engine implementing a predefined cache coherence protocol; and

10 a plurality of input/output nodes, coupled to the interconnect, each input/output node

11 including:

12 no processor cores;

13 an input/output interface for interfacing to an input/output bus or input/output
14 device;

15 a memory cache for caching memory lines of information;

16 an interface to a local memory subsystem, the local memory subsystem storing
17 a multiplicity of memory lines of information; and

18 a protocol engine implementing the predefined cache coherence protocol.

1 2. The system of claim 1, wherein

2 the protocol engine of each of the processor nodes enables the processor cores therein
3 to access memory lines of information stored in the local memory subsystem and memory
4 lines of information stored in the memory cache of any of the processor nodes and
5 input/output nodes, and maintains cache coherence between memory lines of information
6 cached in the memory caches of the processor nodes and memory lines of information cached
7 in the memory caches of the input/output nodes; and

8 the protocol engine of each of the input/output nodes enables an input/output device
9 coupled to the input/output interface of the input/output node to access memory lines of
10 information stored in the local memory subsystem and memory lines of information stored in
11 the memory cache of any of the processor nodes and input/output nodes, and maintains cache

12 coherence between memory lines of information cached in the memory caches of the
13 processor nodes and memory lines of information cached in the memory caches of the
14 input/output nodes.

1 3. The system of claim 1, wherein the system is reconfigurable so as to include any ratio
2 of processor node to input/output nodes so long as a total number of processor nodes and
3 input/output nodes does not exceed a predefined maximum number of nodes.

1 4. The system of claim 1, wherein the protocol engine of each of the processor nodes is
2 functionally identical to the protocol engine of each of the input/output nodes.

1 5. The system of claim 1, wherein the protocol engine of each of the processor nodes and
2 the protocol engine of each of the input/output nodes includes:

3 a memory transaction array for storing an entry related to a memory
4 transaction, the entry including a memory transaction state, the memory transaction
5 concerning a memory line of information; and

6 logic for processing the memory transaction, including advancing the memory
7 transaction when predefined criteria are satisfied and storing a state of the memory
8 transaction in the memory transaction array.

1 6. The system of claim 5, wherein the protocol engine of each of the processor nodes and
2 the protocol engine of each of the input/output nodes is configured to add an entry related to a
3 memory transaction in the memory transaction array in response to receipt by the protocol
4 engine of a protocol message related to the memory transaction.

1 7. The system of claim 1, wherein
2 the processor nodes and the input/output nodes collectively comprise nodes of the
3 system;

4 each node of the processor nodes and the input/output nodes includes:

5 a directory including a respective entry associated with each respective
6 memory line of information stored in the local memory subsystem of the node, the entry

7 including an identification field for identifying a subset of the system nodes caching the
8 memory line of information; and
9 the protocol engine of each of the processor nodes and the protocol engine of each of
10 the input/output nodes includes logic for:
11 configuring the identification field of each directory entry to comprise a
12 plurality of bits at associated positions within the identification field;
13 associating with each respective bit of the identification field one or more
14 nodes of the plurality of nodes, including a respective first node, wherein the one or more
15 nodes associated with each respective bit are determined by reference to the position of the
16 respective bit within the identification field;
17 setting each bit in the identification field of the directory entry associated with
18 the memory line for which the memory line is cached in at least one of the associated nodes;
19 and
20 sending an initial invalidation request to no more than a first predefined
21 number of the nodes associated with set bits in the identification field of the directory entry
22 associated with the memory line.

1 8. The system of claim 1, wherein

2 the processor nodes and the input/output nodes collectively comprise nodes of the
3 system;

4 each node of the processor nodes and the input/output nodes includes:

5 input logic for receiving a first invalidation request, the invalidation request
6 identifying a memory line of information and including a pattern of bits for identifying a
7 subset of the plurality of nodes that potentially store cached copies of the identified memory
8 line; and

9 processing circuitry, responsive to receipt of the first invalidation request, for
10 determining a next node identified by the pattern of bits in the invalidation request and for
11 sending to the next node, if any, a second invalidation request corresponding to the first
12 invalidation request, and for invalidating a cached copy of the identified memory line, if any,
13 in the particular node of the processor computer system.

1 9. A computer system, comprising:

2 a plurality of multiprocessor nodes, each multiprocessor node including:

3 a multiplicity of processor cores, each processor core having an associated
4 memory cache for caching memory lines of information;

5 an interface to a local memory subsystem, the local memory subsystem storing
6 a multiplicity of memory lines of information; and

7 a protocol engine implementing a predefined cache coherence protocol; and

8 a plurality of input/output nodes, each input/output node including:

9 no processor cores;

10 an input/output interface for interfacing to an input/output bus or input/output
11 device;

12 a memory cache for caching memory lines of information;

13 an interface to a local memory subsystem, the local memory subsystem storing
14 a multiplicity of memory lines of information; and

15 a protocol engine implementing the predefined cache coherence protocol.

1 10. The system of claim 9, wherein

2 the protocol engine of each of the multiprocessor nodes enables the processor cores
3 therein to access memory lines of information stored in the local memory subsystem and
4 memory lines of information stored in the memory cache of any of the multiprocessor nodes
5 and input/output nodes, and maintains cache coherence between memory lines of information
6 cached in the memory caches of the multiprocessor nodes and memory lines of information
7 cached in the memory caches of the input/output nodes; and

8 the protocol engine of each of the input/output nodes enables an input/output device
9 coupled to the input/output interface of the input/output node to access memory lines of
10 information stored in the local memory subsystem and memory lines of information stored in
11 the memory cache of any of the multiprocessor nodes and input/output nodes, and maintains
12 cache coherence between memory lines of information cached in the memory caches of the
13 multiprocessor nodes and memory lines of information cached in the memory caches of the
14 input/output nodes.

1 11. The system of claim 9, wherein the system is reconfigurable so as to include any ratio
2 of multiprocessor node to input/output nodes so long as a total number of multiprocessor
3 nodes and input/output nodes does not exceed a predefined maximum number of nodes.

1 12. The system of claim 9, wherein the protocol engine of each of the multiprocessor
2 nodes is functionally identical to the protocol engine of each of the input/output nodes.

1 13. The system of claim 9, wherein the protocol engine of each of the multiprocessor
2 nodes and the protocol engine of each of the input/output nodes includes:

3 a memory transaction array for storing an entry related to a memory
4 transaction, the entry including a memory transaction state, the memory transaction
5 concerning a memory line of information; and

6 logic for processing the memory transaction, including advancing the memory
7 transaction when predefined criteria are satisfied and storing a state of the memory
8 transaction in the memory transaction array.

1 14. The system of claim 13, wherein the protocol engine of each of the multiprocessor
2 nodes and the protocol engine of each of the input/output nodes is configured to add an entry
3 related to a memory transaction in the memory transaction array in response to receipt by the
4 protocol engine of a protocol message related to the memory transaction.

1 15. The system of claim 9, wherein

2 the multiprocessor nodes and the input/output nodes collectively comprise nodes of
3 the system;

4 each node of the multiprocessor nodes and the input/output nodes includes:

5 a directory including a respective entry associated with each respective
6 memory line of information stored in the local memory subsystem of the node, the entry
7 including an identification field for identifying a subset of the system nodes caching the
8 memory line of information; and

9 the protocol engine of each of the multiprocessor nodes and the protocol engine of
10 each of the input/output nodes includes logic for:

11 configuring the identification field of each directory entry to comprise a
12 plurality of bits at associated positions within the identification field;
13 associating with each respective bit of the identification field one or more
14 nodes of the plurality of nodes, including a respective first node, wherein the one or more
15 nodes associated with each respective bit are determined by reference to the position of the
16 respective bit within the identification field;
17 setting each bit in the identification field of the directory entry associated with
18 the memory line for which the memory line is cached in at least one of the associated nodes;
19 and
20 sending an initial invalidation request to no more than a first predefined
21 number of the nodes associated with set bits in the identification field of the directory entry
22 associated with the memory line.

1 16. The system of claim 9, wherein
2 the multiprocessor nodes and the input/output nodes collectively comprise nodes of
3 the system;
4 each node of the multiprocessor nodes and the input/output nodes includes:
5 input logic for receiving a first invalidation request, the invalidation request
6 identifying a memory line of information and including a pattern of bits for identifying a
7 subset of the plurality of nodes that potentially store cached copies of the identified memory
8 line; and
9 processing circuitry, responsive to receipt of the first invalidation request, for
10 determining a next node identified by the pattern of bits in the invalidation request and for
11 sending to the next node, if any, a second invalidation request corresponding to the first
12 invalidation request, and for invalidating a cached copy of the identified memory line, if any,
13 in the particular node of the multiprocessor computer system.